CS 361 - Homework 01 - Theory

Professor Mark W. Boady

1 Overview

This homework consists of a few theory questions. The focus is on possible scenarios of execution for atomic statements.

All answers must be **typed** and submitted as a single **PDF**. The PDF must include your name and all questions must be clearly labeled. You do not have to repeat the question text, only label the questions clearly.

For all algorithm examples, lines that are not given numbers are considered starting conditions and not part of the algorithm. For example, if an algorithm has a local variable that starts at value $x = 10$, this might not be given a line number. It is a started condition for the threads. In practice, it would be done betore the threads starts.

All lines are assumed to be **atomic statements**. Execution will never context switch in the middle of completing a numbered line.

2 Questions

This homework is worth 50 points.

Question 1 : 20 points

Use the following algorithm to answer this question. The statement `for i from a to b` means i iterates through all values in the range including a and b. In python this would be **for** i **in range**(a,b+1) and in C it would be **for**(**int** i=a; i <= b; i++;). The loop exits when $i = b + 1$.

Algorithm: Concurrent Counting	
Setup: $n = 0$	
Thread P	**Thread Q**
local variable temp, i	local variable temp, j
p1: for i from 1 to 10 do	q1: for j from 1 to 10 do
p2: temp=n	q2: temp=n
p3: n=temp+1	q3: n=temp+1

Note: If your scenario table has extended repetitions you may write a row stating something like "Previous 3 lines repeat 4 times" instead of duplicating the same sequence.

(a) (10 points) Give a Scenario where the final value of n is 10. Show the Scenario Table.

(b) (10 points) Give a Scenario where the final value of n is 2. Show the Scenario Table.

Question 2 : 30 points

Three algorithms that all solve the same problem are given below.

Assume that for the function f, there is one unique integer value $i \in \mathbb{Z}$ for which $f(i) = 0$. Each of the following algorithms searches for i.

Each of the algorithms does not work as intended.

Provide one counterexample to show why the algorithm does not work correctly. To work correctly, **both** threads **must** exit and it **must** be obvious what the correct value of i is.

State any assumptions before the start of the scenario table. For example, you might say something like "assume Thread P has found f(i)==0 to be true on line p4." Then start the scenario table on line p4. Your scenario table should focus on showing the flaw of the algorithm happening.

(a) (10 points) Write a Scenario Table showing this algorithm does not work.

Algorithm Zero A	
Setup: global boolean found	
Thread P	**Thread Q**
local variable i=0	local variable j=1
p1: found = false	q1: found = false
p2: while not found	q2: while not found
p3: i=i+1	q3: j=j-1
p4: found = (f(i)==0)	q4: found = (f(j)==0)

(b) (10 points) Write a Scenario Table showing this algorithm does not work.

Algorithm Zero B	
Setup: boolean found=false	
Thread P	**Thread Q**
local variable i=0	local variable j=1
p1: while not found	q1: while not found
p2: i=i+1	q2: j=j-1
p3: found = (f(i)==0)	q3: found = (f(j)==0)

(c) (10 points) Write a Scenario Table showing this algorithm does not work.

Algorithm Zero C	
Setup: boolean found=false, turn=1	
Thread P	**Thread Q**
local variable i=0	local variable j=1
p1: while not found	q1: while not found
p2: await turn==1 then turn=2	q2: await turn==2 then turn=1
p3: i=i+1	q3: j=j-1
p4: if f(i)==0	q4: if f(j)==0
p5: found = true	q5: found = true